**计算机程序设计基础（C++)**

**实验报告**

专业班级：软件工程2301

学 号：8209230102

姓 名：刘喜贺

**实验报告成绩：**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验** | **实验一** | **实验二** | **实验三** | **实验四** | **实验五** | **总评** |
| **成绩** |  |  |  |  |  |  |

**批阅教师：**

**实验一、实验环境与简单程序设计**

**一、实验目的**

1、掌握集成开发环境，掌握C++程序的基本要素以及完整的C++程序开发过程。

2、掌握基本数据类型、运算符和表达式的使用。理解隐式转换和强制转换，理解数据超过该数据类型

表示范围时的溢出。掌握不同数据之间的混合算术运算中数据类型的转换。

3、变量的定义与常量的使用。

4、输入、输出的实现。

5、编译信息的理解与错误的修改。

6、简单程序的设计。

**二、实验内容**

熟悉C++编程环境，可以使用VS；对已经能熟练掌握C++开发环境的同学，可

以跳过本部分内容）

**1.编辑输入下列程序，找出下面代码的错误并改正：**

#include<iostream>

using namespace std;

int main()

{

Int k = 1;

Int i = k + 1;

cout << ~~i++~~ ++i<< endl;

~~int~~ i = 1;

cout << ~~i++~~++i << endl;

cout << “Welcome to C++”<<endl;

return 0;

}

**2.求圆锥的体积：要求键盘输入圆锥底的半径、锥高，使用标识符常量定义圆周率。**

(1)创建一个控制台项目

(2)在文件中输入程序内容，存盘

(3)编译、连接、运行；观察结果

**3**.**通过下面程序验证你所使用系统上运行的C++编译器中每个基本数据类型的长度。**

#include<iostream>

using namespace std;

int main()

{

cout << “char length:” << sizeof(char) << endl;

cout << “int length:” << sizeof(int) << endl;

}

**4.观察下面程序的执行结果。**

#include <iostream>

#include <iomanip>

using namespace std;

int main()

unsigned int testUnint=65534;//oxfffe

cout << “output in unsigned int 1 type:”” << testUnint<< end;//<<oct;

cout << “output in char type:!” << static\_ cast<char>(testUnint)<< endl;

cout << “output in short type:” << static\_ cast<short>(testUnint)<< endl;//为什么结果为-2?:

cout << “output in int type:” << static\_ cast<int>. (testUnint)<< endl;

cout << “output in double type:”<< static cast<double>(testUnint)<< endl;

cout << “output in double type:” <<setprecision(4)<< static\_ cast<double>(testUnint)<< endl;

cout << “output in Hex unsigned int type:” <<hex<< testUnint<< endl; //16进制输出

system(“pause”);

return 0;

**自己编程测试一下将testUnint按8进制输出<<oct；将一个实数转换成int,观察结果。**

**5.编程，输入华氏温度，将其转换为摄氏温度后输出（保留两位小数）。**

**三、算法分析，程序结果**

1.

#include<iostream>

using namespace std;

int main()

{

//int i = k + 1; 没有初始化变量k 无法赋值

int k = 1;

int i = k + 1;

//cout << i++ << endl; i++是先输出i的原始值 再对i进行+1的操作 在此步中 i++没有意义

cout << ++i << endl;

//int i = 1; i重定义 报错

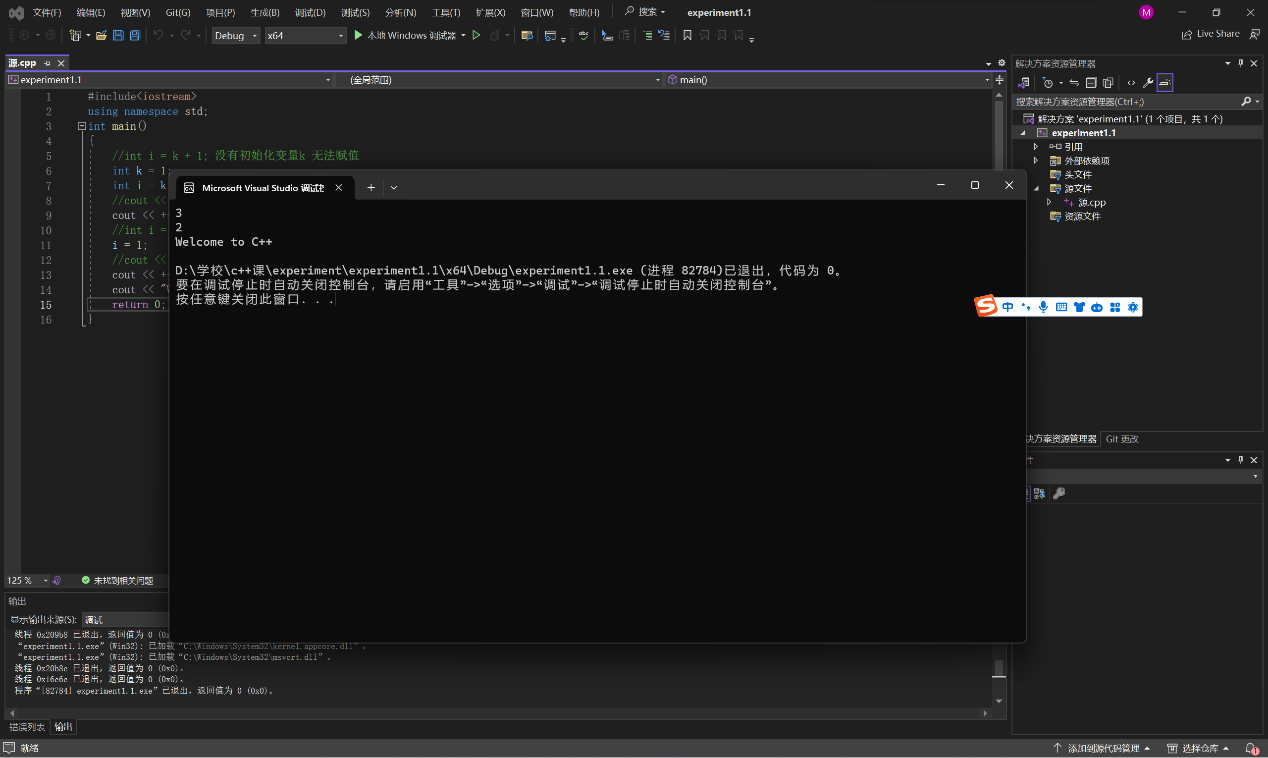
i = 1;

//cout << i++ << endl; i++是先输出i的原始值 再对i进行+1的操作 在此步中 i++没有意义

cout << ++i << endl;

cout << “Welcome to C++” << endl;

return 0;

}

2.

#include<iostream>

#define π 3.14;//定义宏常量 防止值被改

using namespace std;

int main()

{

double r;//定义半径

double h;//定义高度

cout << “请输入圆锥底面半径:”;//提示输入半径

cin >> r;//输入半径

cout << “请输入圆锥高:”;//提示输入高

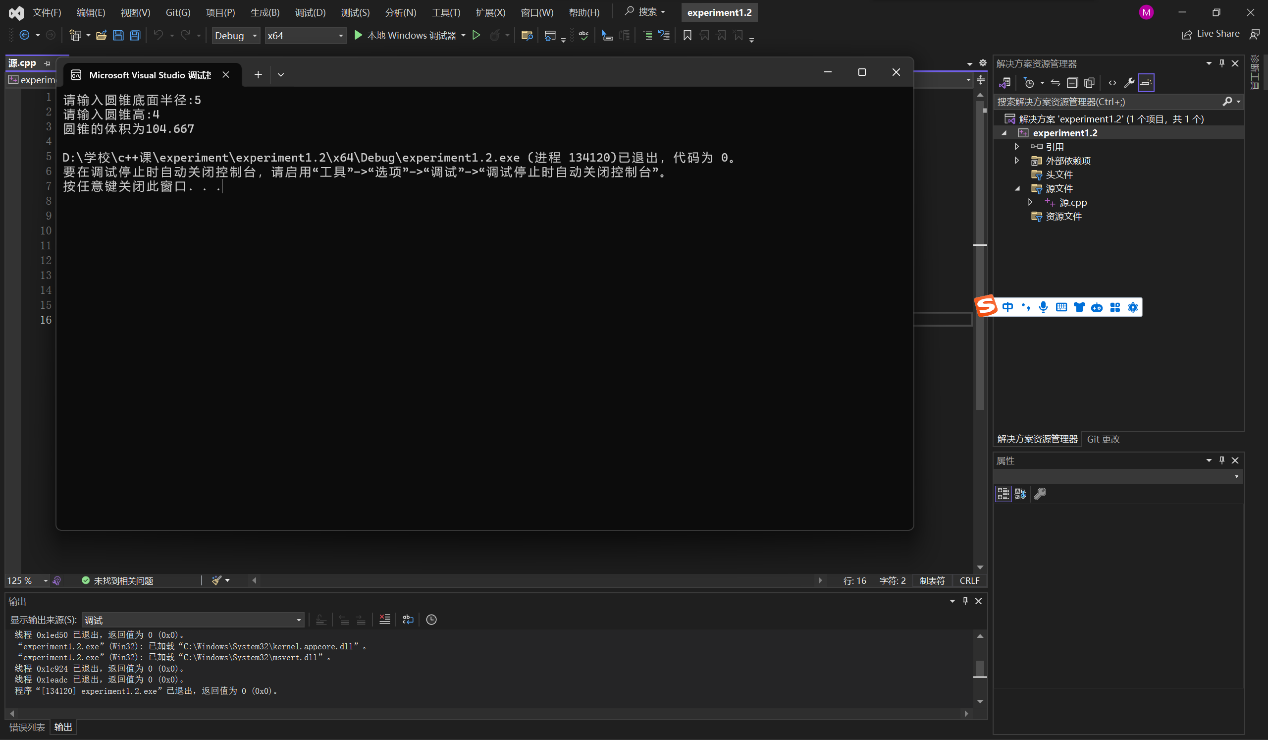
cin >> h;//输入高

double s = (r \* r) \* π;//s=πr^2 定义s并给其赋值

double v = s \* h \* (1.0 / 3);//v=1/3sh 定义v并给其赋值

cout << “圆锥的体积为” << v << endl;

return 0;

}

3.

#include<iostream>

using namespace std;

int main()

{

cout << “char length:” << sizeof(char) << endl;//输出char类型数据的长度

cout << “int length:” << sizeof(int) << endl;//输出int类型数据的长度

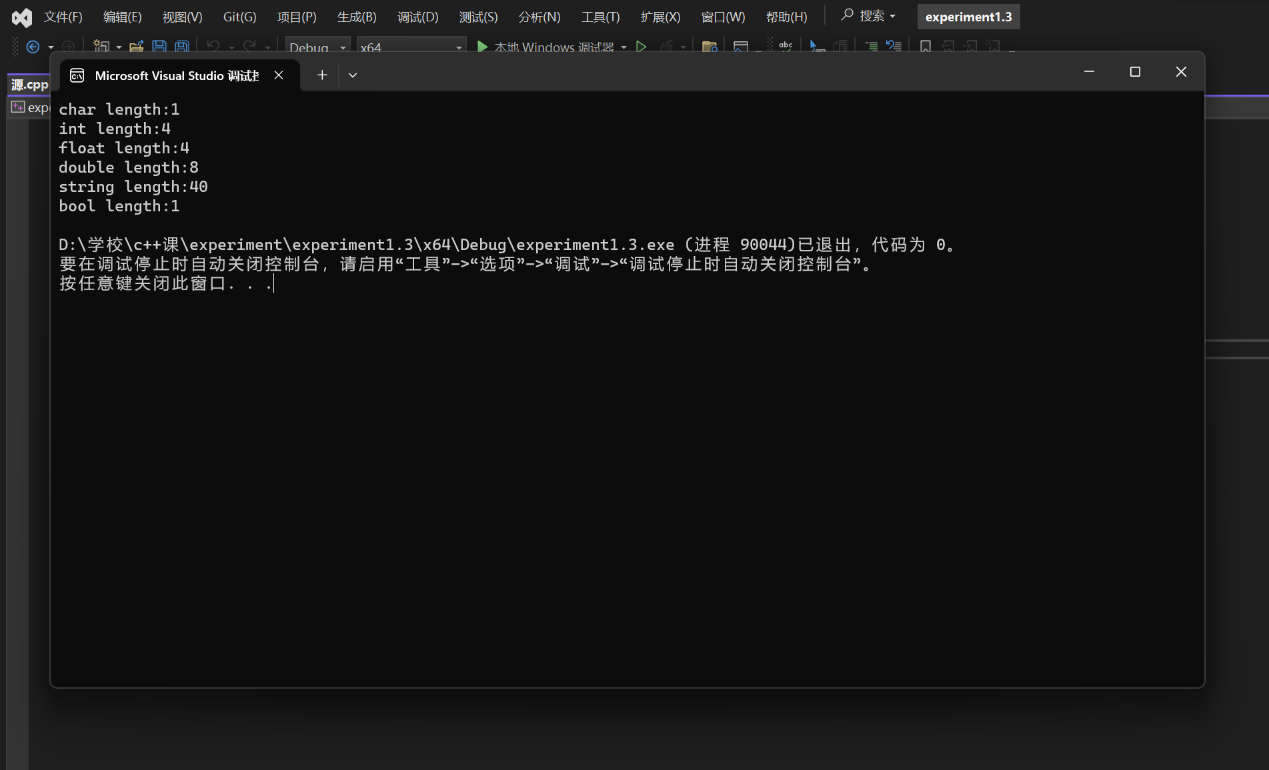
cout << “float length:” << sizeof(float) << endl;//输出float类型数据的长度

cout << “double length:” << sizeof(double) << endl;//输出double类型数据的长度

cout << “string length:” << sizeof(string) << endl;//输出string类型数据的长度

cout << “bool length:” << sizeof(bool) << endl;//输出bool类型数据的长度

return 0;

}

4.

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

unsigned int testUnint = 65534;//oxfffe

cout << “output in unsigned int 1 type:” << testUnint << endl;//<<oct;

cout << “output in char type:!” << static\_cast<char>(testUnint) << endl;

cout << “output in short type:” << static\_cast<short>(testUnint) << endl;//为什么结果为-2?:

cout << “output in int type:” << static\_cast<int>(testUnint) << endl;

cout << “output in double type:” << static\_cast<double>(testUnint) << endl;

cout << “output in double type:” << setprecision(4) << static\_cast<double>(testUnint) << endl;

cout << “output in Hex unsigned int type:” << hex << testUnint << endl; //16进制输出

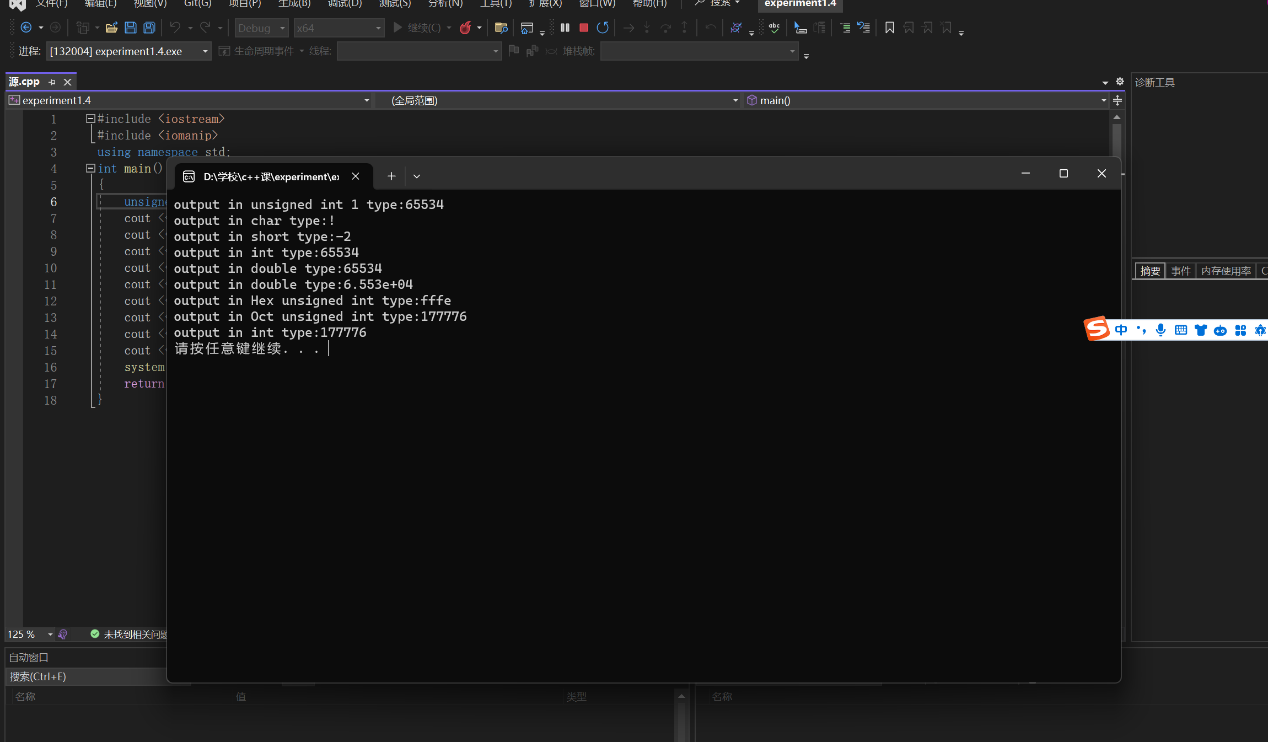
cout << “output in Oct unsigned int type:” << oct << testUnint << endl;//八进制输出

cout << “output in int type:” << static\_cast<int>(testUnint) << endl;//转换成int

system(“pause”);

return 0;

}



5.

#include<iostream>

#include<iomanip>//用setprecision要引用该头文件

using namespace std;

int main()

{

double F;//定于华氏温度变量

cout << “请输入华氏温度:”;//提示输入华氏温度

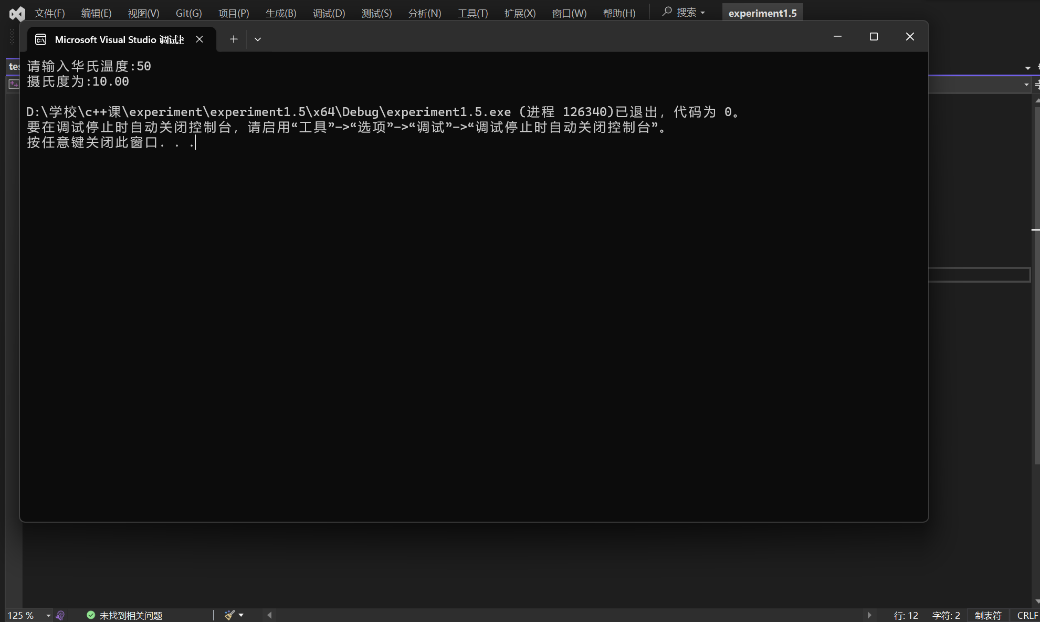
cin >> F;//输入华氏温度

double C = (F – 32) / 1.8;//定义摄氏温度 计算并赋值

cout << “摄氏度为:” <<fixed << setprecision(2) << C << endl;//输出摄氏度 使用fixed和setprecision保留两位小数

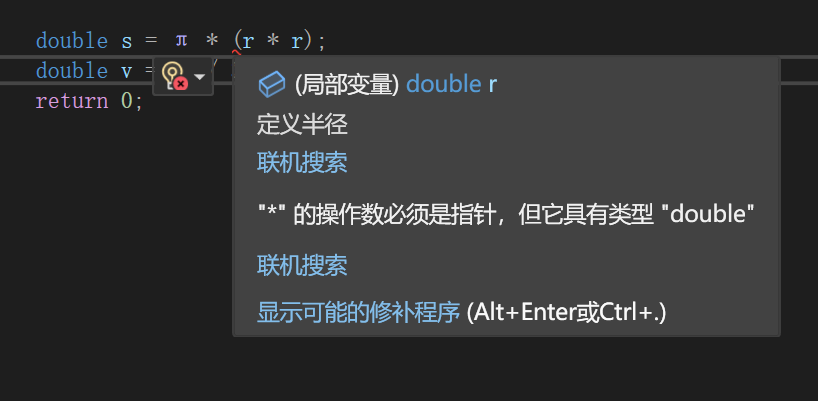
return 0;

}

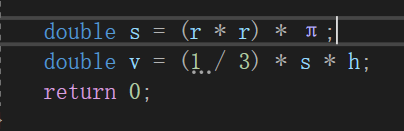


**四、遇到的问题与解决方法**

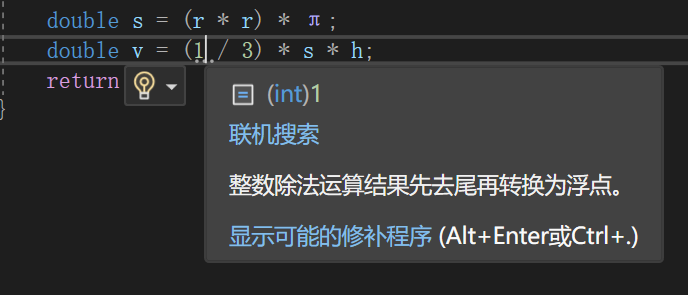
**1.在计算圆锥体积时，遇到报错及提示**

 （1）报错

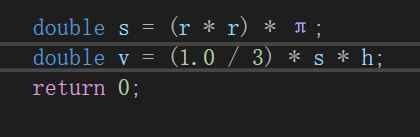
根据编译器的报错提示，推测是由于在r变量前加了\*号，使编译器识别成了指针变量。

 考虑到r是局部变量，将（r\*r）提前写，后面写“\*π”再次尝试。

报错消除，程序能够正常运行

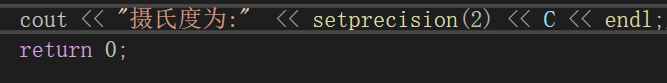
（2）提示

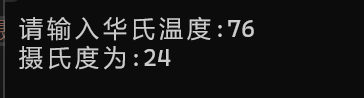
在计算1/3分数时获得编译器提示，想起整数除法运算会去尾，影响计算准确度。

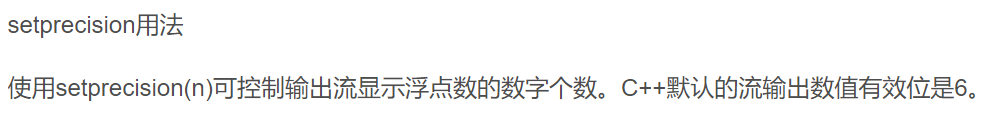
 考虑将整数转换为小数，改为1.0/3，提示消除，程序能够正常运行，计算准确度提高。

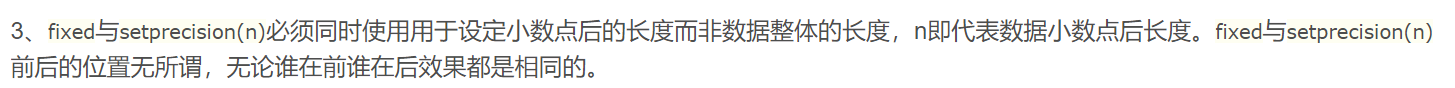
**2.不知道如何保留两位小数**

在编辑华氏度转摄氏度的程序时，体重要求结果保留两位小数。

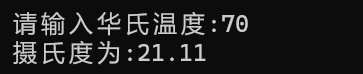
 首先考虑通过setprecision设置显示浮点型数据的位数。

 在设置显示位数后，运行程序，发现运行结果不佳。

 随后查阅相关资料，想起单使用setprecision控制的是总的显示位数，不控制小数点后的显示位数

 再次查询相关资料，发现要想保留小数点两位，需要fixed与setprecision结合使用。

 遂在代码中加入fixed。

 发现代码可以正常运行，两位小数可以正常保留，代码修改成功。

**五、体会**

**1.在日常C++的学习中，掌握实践技能固然重要，但不能遗忘一些重要的理论知识。**

在此次实验一的实验过程中，存在一些对于代码底层逻辑以及代码运算规则的考察。在编程的过程中，发现我可能更擅长代码的应用编写，而在面对底层逻辑运算规则的考察时，需要考编译器提供提示，才能注意到自己犯错的点。

例如，在实验1.1当中，出现了两次对于变量i的定义。我在最开始还没有将这段代码录进编译器时，没有注意到这一点错误。直到将此段代码在编译器中打出，发现编译器在此处提示报错，在查看编译器的报错提示后，才发现在这段代码中存在着对于变量i的重复定义。可能在日常的编程中，自己比较少遇到这一类错误，所以无法立即发现题中的这一点问题。在看到编译器的提示后才对原代码中的这一个问题恍然大悟。希望这一次经历能够加深我对这一类编程语法的重视性。

再例如，在计算圆锥的体积是，通过公式V=1/3sh计算。在输入代码时，直接打上的是V=s\*h\*(1/3)，并没有发现错误点。在看到编译器提示后，才想起整数除法会干扰数据结果的准确度，将（1/3）改为（1.0/3）。其实在上课时，老师在介绍运算原理时，解释了很多相关原理及方式，但是感觉自己对于这方面还没有完全掌握。希望这一次经历能够加深我对这一类知识点的理解。

**2.在学习了后续的内容后，容易遗忘前面学到的知识，需要多加复习**

实验一涉及到的c++编程知识都是很基础的内容。在进行编程的实际过程中，我发现，对于前期学到的、一些比较少用的知识点会出现遗忘的情况。需要借助相关资料，才能回想起某一个知识点的所有细节内容。

例如，在试图控制浮点型数据显示位数的时候，回想起有一种方式可以调整位数，也会记得使用该方法需要引入头文件。但是无法仔细想起如何拼写该函数，也无法想起要引入哪一个头文件。只能通过通过查找相关资料的方法，回想起此函数的使用方式。

再例如，在进行进制转换时，也对进制转换的方式有所遗忘。在老师讲述这方面内容时，对这块知识掌握程度比较高，但是时隔长时间后，已经全然忘记如何进行进制转换进制转换，需要参考原代码中给出的方式才能成功实现进制转换。

实验一的实验帮助我对一些知识点重新进行了复习，也让我认识到了复习旧知识的重要性。

**3.在进行自学时，不能只看重速度，还要注意质量**

在平日里进行自学时，我的学习目标可能是运用这一方面的知识，不会很对知识原理分散过多的注意力，这可能导致学习质量不是很高。

在输入公式s=π\*(r\*r)时，发现编译器将\*（r\*r）理解为了指针。虽然我可以立即将次报错取消，但是在取消报错之后我也无法理解为何编译器会在此处报错。可能是因为当初在进行指针相关知识的学习过程中，我对知识的理解程度不是很到位，例如，我对于“int \*p=1；”定义后，有时会难以分清“\*p”以及“p”的区别，有时又可以分清。

所以在这一实验过程中，我明白了自学过程中质量的重要性。

**实验二、数据结构**

**一、实验目的**

1、学习与掌握逻辑运算与逻辑表达式。

2、熟练掌握if、switch、while、do-while，for语句的语法结构与执行过程。

3、掌握选择、循环程序的设计方法

**二、实验内容**

1、输入一个字符，如果为小写，转换为大写输出，否则，输出其后继字符的ASCII码值。

2、输入x计算表达式的值：



分别输入 0.2, 1, 5 , 0,观察输出结果。

1. 输入三角形的三条边，求周长，并判断该三角形是否为等腰三角形（提示：要三边是否可以构成三角形）。

4、 完成计算器程序，实现（+ - \* / %）运算。考虑除数为 0 与运算符非法的情况。

5、输入一行字符，分别统计出其中英文字母、空格、数字字符和其它字符的个数。

提示：从键盘上读入一个字符给变量 c，判断 c 是属于哪种字符并计数，循环读入下个字符，直到回车换行字符'\n'为止。

cin，scanf（）都不能读入空格以及‘\n’字符，查找资料解决输入这两个字符的方法。

（这个题训练大家自主学习能力以及如何获取新知识、探索解决未知问题的能力。）

6、编写一个程序：从键盘上输入两个正整数，求 a 和 b 的最大公约数与最小公倍数。

7、使用循环结构输出下列图形：

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

8、从键盘输入 a，用迭代法求 a 的平方根 x= *a* 。求平方根的迭代公式为：

要求精确到|xn+1 - xn|<10 -5。

提示：迭代法是把 xn代入迭代公式右边，计算出 xn+1来，然后把 xn+1 作为新的 xn ，计算出新的 xn+1，如此重复，直到|xn+1 - xn|<10 -5 时，xn+1 为所求的平方根。可以把 a 作为 xn 的初始值。

思考：（1）如果输入 a 为负，在运行时会出现什么情况? 修改程序使之能处理任何的 a 值。

(2）能否|xn+1 -xn|<10 -10或更小? 为什么? 请试一下。 可以

9、苹果每个 0.8 元，第一天买 2 个，第二天开始，每天买前天的 2 倍，直到购买的苹果数不超过100的最大值，求每天平均花多少钱。

**三、算法分析，程序结果**

1.

#include<iostream>

using namespace std;

int main()

{

char ch;//定义ch

cout << "请输入一个字符" << endl;//提示输入

cin >> ch;//输入

int a = int(ch);//定义a 并将ASCII码赋值给a

if (a >= 97 && a <= 122)//判断 如果ASCII码在小写字母之间

{

a -= 32;//ASCII码-32 变为大写

cout << char(a);//输出大写后的字符

}

else//没有大写 或者已经为大写

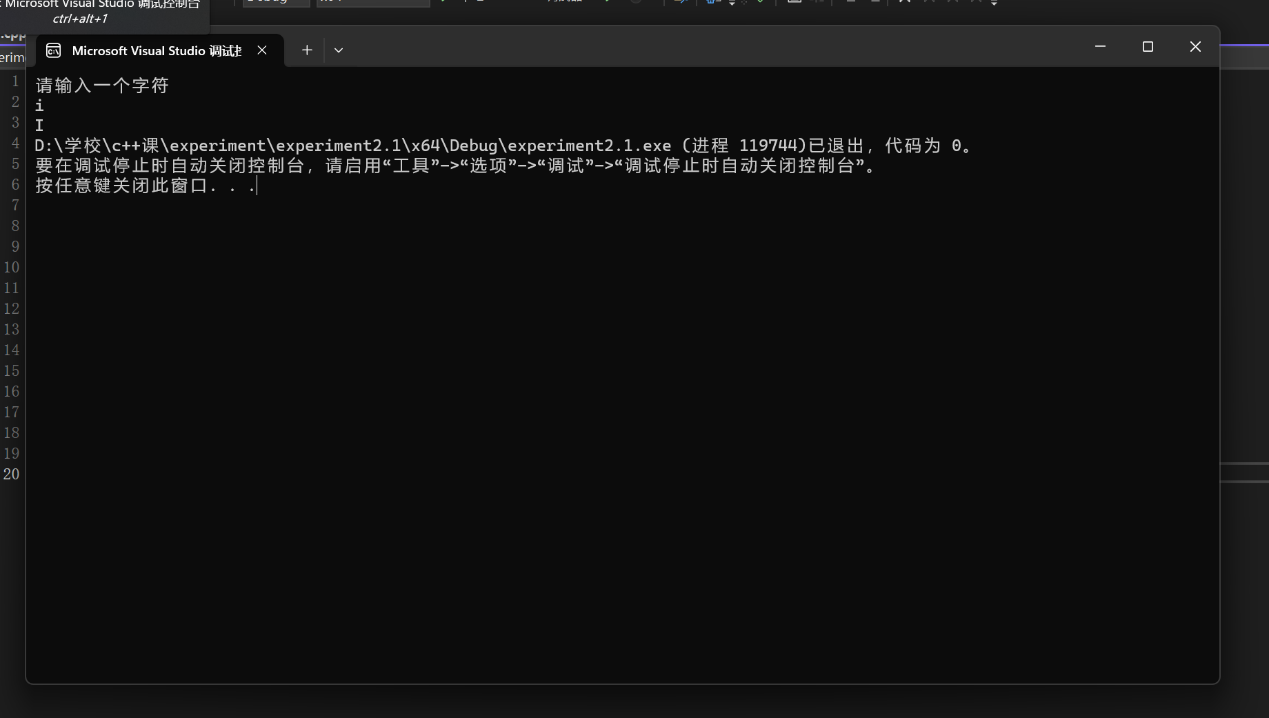
{

a++;//ASCII码+1

cout << a;//输出+1后的值

}

return 0;

}

2.

#include<iostream>

using namespace std;

int main()

{

double x;//定义自变量x

double y;//定义因变量y

cout << "请输入x的值:";//提示输入x的值

cin >> x;//输入x的值

if (x > 0 && x < 1)//如果0<x<1

{

y = 3 - (2 \* x);//计算出y的值

cout << "y=" << y << endl;//输出y的值

}

else if (x >= 1 && x < 5)//如果1<=x<5

{

y = 2 / (4 \* x) + 1;//计算出y的值

cout << "y=" << y << endl;//输出y的值

}

else if (x >= 5 && x < 10)//如果5=<x<10

{

y = x \* x;//计算出y的值

cout << "y=" << y << endl;//输出y的值

}

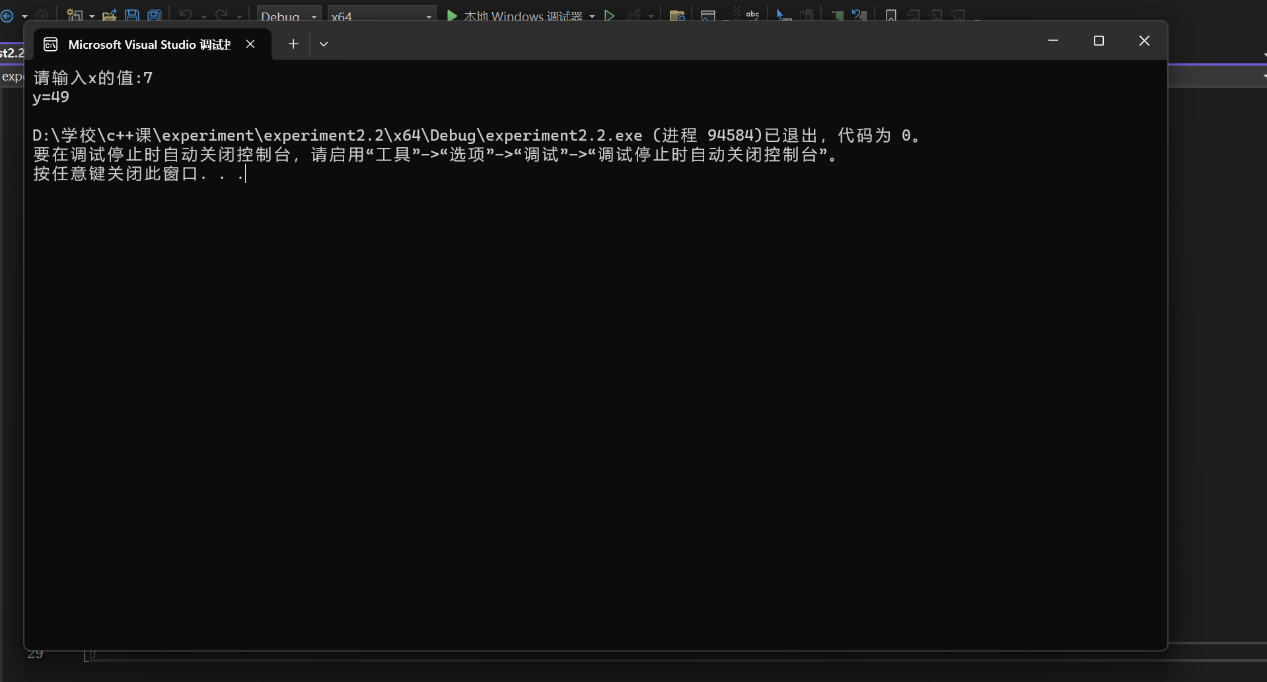
else//都不满足

{

cout << "您输入的x值不在定义域内" << endl;//不在定义域里

}

return 0;

}

3.

#include<iostream>

using namespace std;

int main()

{

int s1;//定义边长1

int s2;//定义边长2

int s3;//定义边长3

cout << "请分别输入三边边长" << endl;//提示输入三条边边长

cout << "边长1:";//提示输入边长1

cin >> s1;//输入边长1

cout << "边长2:";//提示输入边长2

cin >> s2;//输入边长2

cout << "边长3:";//提示输入边长3

cin >> s3;//输入边长三

if (s1 + s2 > s3 && s1 + s3 > s2 && s2 + s3 > s1)//判断是否满足三角形构成条件(两边之和大于第三边)

{

//满足三角形构成条件

if (s1 == s2 || s2 == s3 || s1 == s3)//判断是否存在两边边长相等的情况

{

cout << "您创建的是等腰三角形" << endl;//输出等腰三角形结果

}

cout << "三角形周长为" << (s1 + s2 + s3) << endl;//输出三角形周长

}

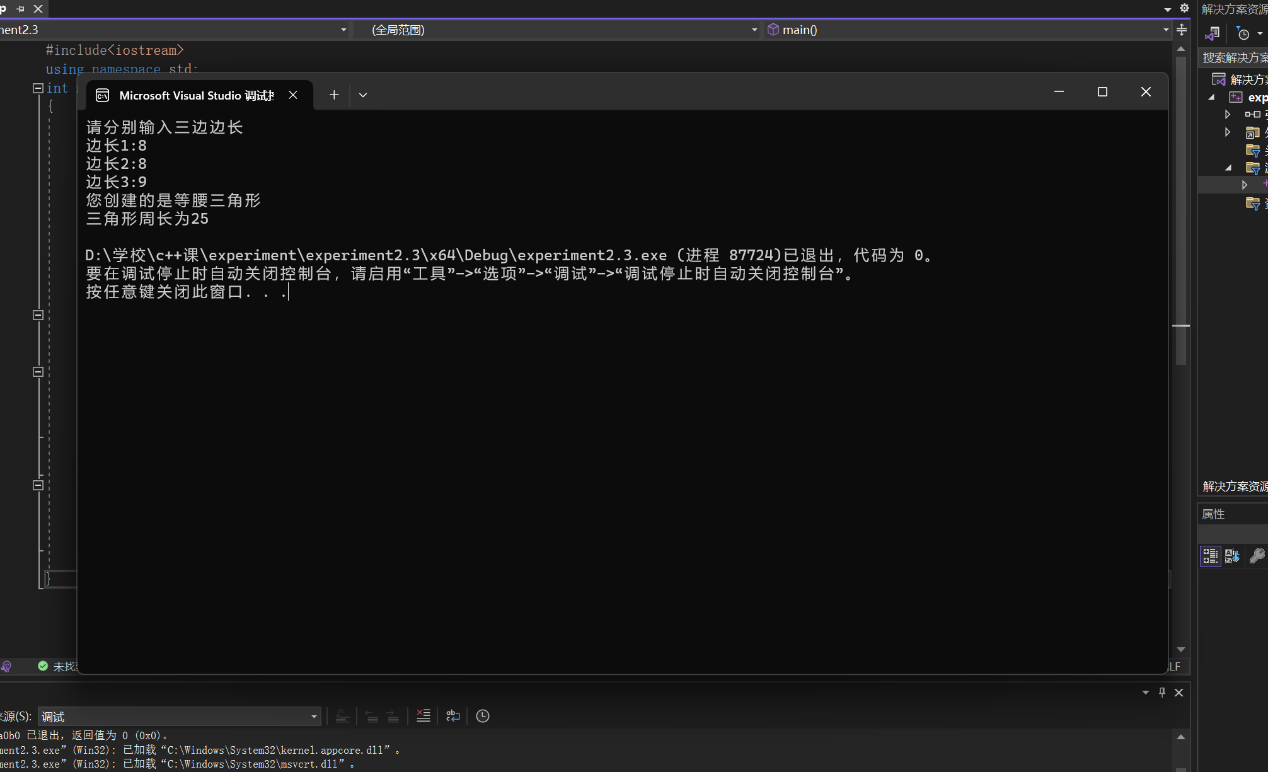
else//不满足三角形构成条件

{

cout << "您输入的三边长不能组成三角形" << endl;//输出提示三角形不能构成

}

return 0;

}

4.

#include<iostream>

using namespace std;

int main()

{

cout << "欢迎使用本计算器系统!" << endl;//提示进入计算器程序

cout << "请输入第一个值:" << endl;//提示输入值1

double num1;//定义值1

cin >> num1;//输入值1

cout << "请选择你要进行的运算" << endl;//提示选择进行的运算(避免了运算符判与选择)

cout << "1.加法运算" << endl;

cout << "2.减法运算" << endl;

cout << "3.乘法运算" << endl;

cout << "4.除法运算" << endl;

cout << "5.取余运算" << endl;

int select;//定义选择值

cin >> select;//输入选择值

cout << "请输入第二个值:" << endl;//提示输入值2

double num2;//定义值2

cin >> num2;//输入值2

switch (select)//判断选择值

{

case 1://1 执行加法运算

cout << "两数之和为" << (num1 + num2) << endl;//输出加法结果

break;

case 2://2 执行减法运算

cout << "两数之差为" << (num1 - num2) << endl;//输出减法结果

break;

case 3://3 执行乘法运算

cout << "两数之积为" << (num1 \* num2) << endl;//输出乘法结果

break;

case 4://4 执行除法运算

if (num2 == 0)//判断除数是否为0

{

//如果除数为0

cout << "除数不能为0" << endl;//提示除数不能为0

break;

}

else//除数不为0

{

cout << "两数之商为" << (num1 / num2) << endl;//输出除法结果

break;

}

case 5://5 执行取余运算

if (num2 == 0)//判断取余数是否为0

{

//如果取余数为0

cout << "除数不能为0" << endl;//提示取余数不能为0

break;

}

else//取余数不为0

{

cout << "两数取余结果为" << (int(num1) % int(num2)) << endl;//输出取余结果

break;

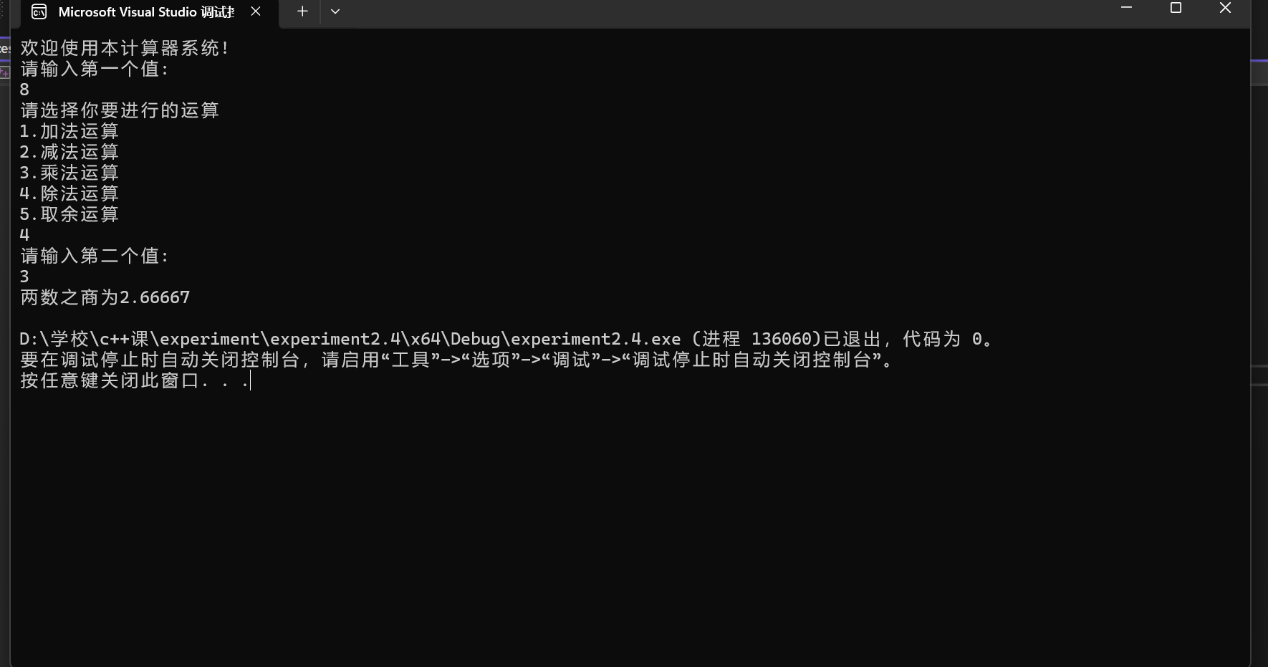
}

default:

cout << "请选择合法运算" << endl;//选择的运算不合法 提示选择合法运算

}

return 0;

}

5.

#include<iostream>

using namespace std;

int main()

{

char c;//定义一个字符

int letter = 0, space = 0, number = 0, other = 0;//定义各种字符类型

cout << "请输入一串字符串" << endl;//提示输入字符串

while ((c=cin.get()) != '\n')//如果cin中不包含换行 循环继续

{

if (c>='a'&&c<='z' || c>='A'&&c<='Z')//判断字母

letter++;//字母字符+1

else if (c==' ')//判断空格

space++;//空格字符+1

else if (c>='0'&&c<='9')//判断数字

number++;//数字字符+1

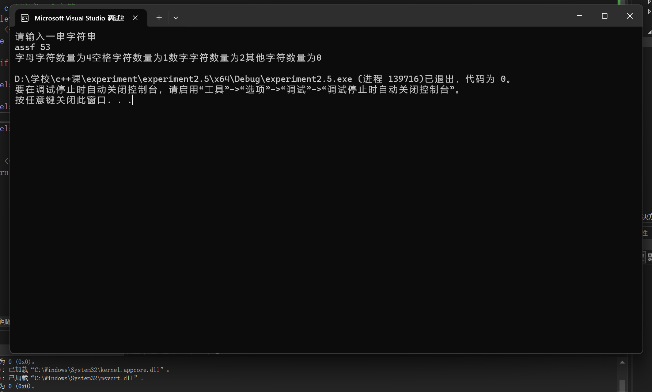
else

other++;//否则其他字符+1

}

cout << "字母字符数量为" << letter << "空格字符数量为" << space << "数字字符数量为" << number << "其他字符数量为" << other << endl;//输出各类字符数量

return 0;

}

6.

#include<iostream>

using namespace std;

int Gcd(int num1, int num2)//定义最大公因数函数

{

int temp;//定义临时变量

for (int i = 1; i < num1 || i < num2; i++)//设置循环

{

if (num1 % i == 0 && num2 % i == 0)//如果i作为a,b的取余数均为0 就都是a,b的因数

{

temp = i;//将目前最大的i覆盖赋值到临时变量

}

}

return temp;//返回临时变量

}

int LCM(int num1, int num2)//定义最小公倍数函数

{

int temp;//定义临时变量

for (int i = num1; i <= (num1 \* num2); i++)//设置循环

{

if (i % num1 == 0 && i % num2 == 0)//如果i作为被取余数,分别取余a,b结果为0 就是a,b的公倍数

{

temp = i;//将公倍数赋值到临时变量

break;//找到的第一个公倍数就是最小公倍数 跳出循环

}

}

return temp;//返回临时变量

}

int main()

{

int a, b;//定义a,b两个变量

cout << "请输入数字a:";//提示输入a

cin >> a;//输入a

cout << "请输入数字b:";//提示输入b

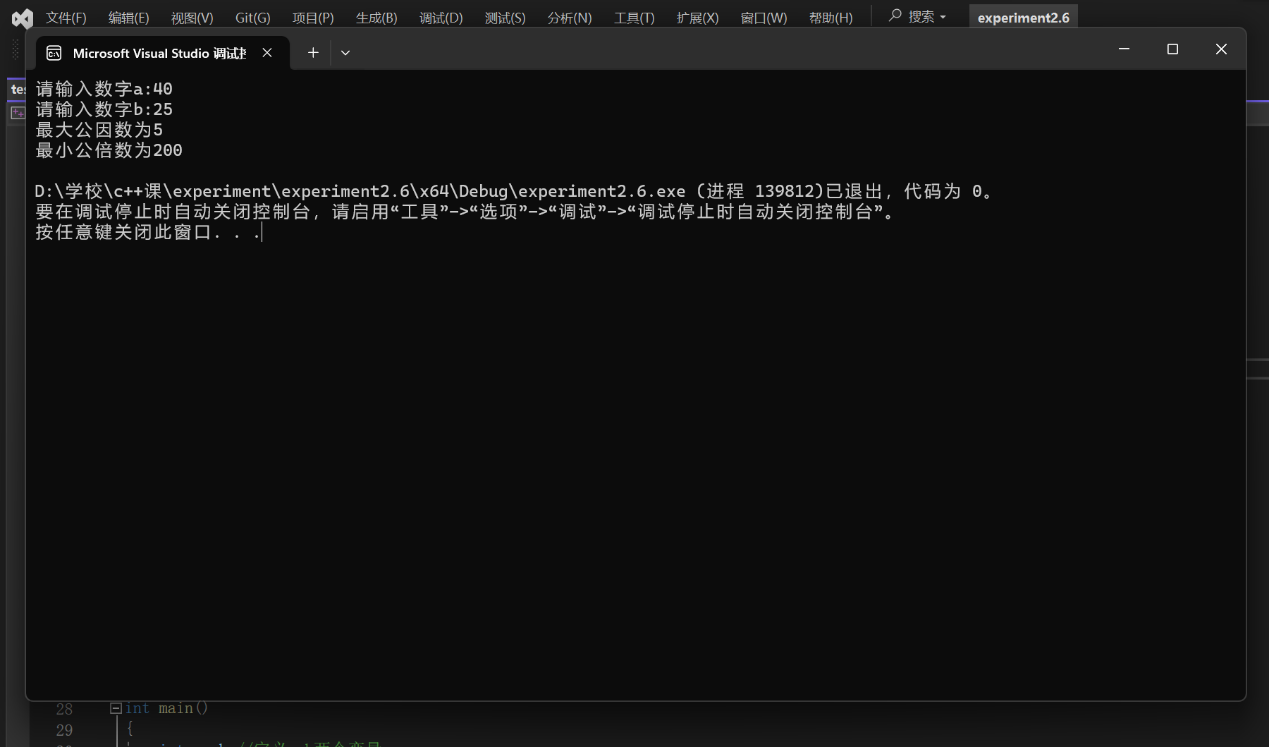
cin >> b;//输入b

int gcd = Gcd(a, b);//调用最大公因数函数

int lcm = LCM(a, b);//调用最小公倍数函数

cout << "最大公因数为" << gcd << endl << "最小公倍数为" << lcm << endl;//输出结果

return 0;

}

7.

#include<iostream>

using namespace std;

int main()

{

for (int i = 1; i < 6; i++)//定义一个循环 i为行数 一共五行 i<6

{

for (int j = 1; j <= i; j++)//定义嵌套循环 j为打印\*数

{

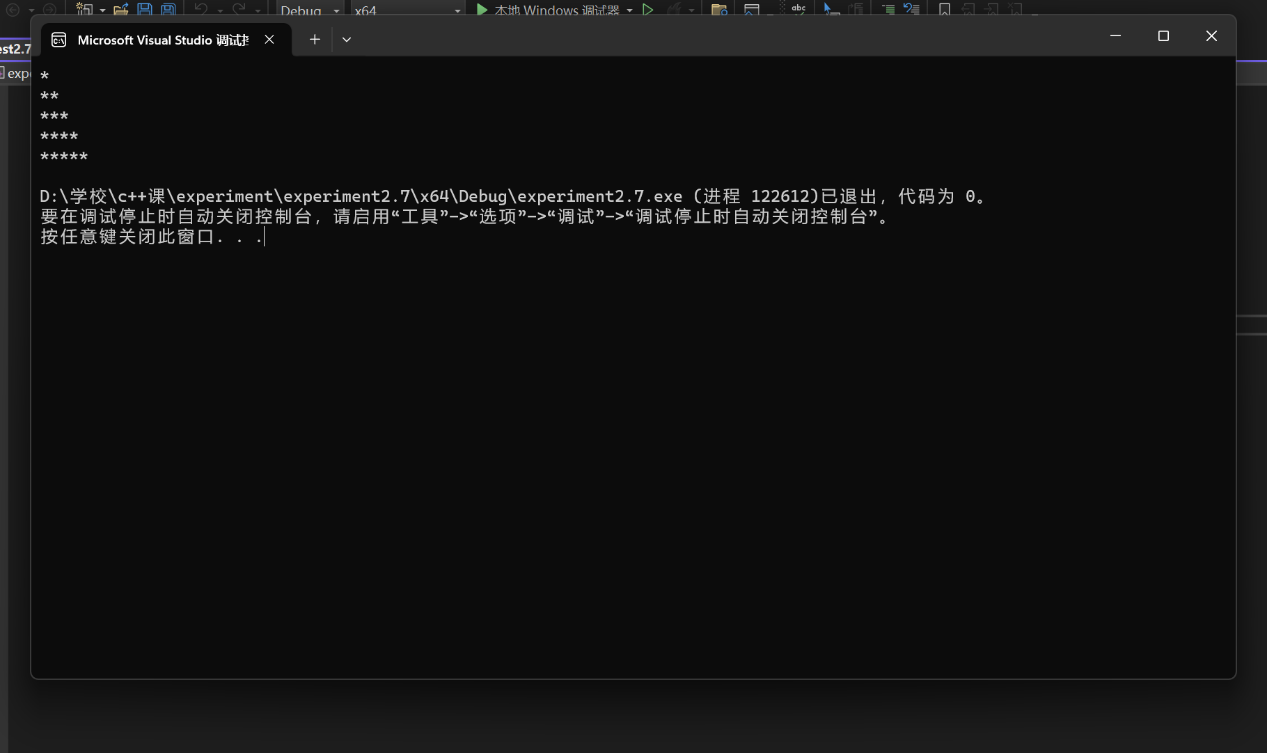
cout << '\*';//打印\*

}

cout << endl;//本行打印完换行

}

return 0;

}

8.

#include<iostream>

#include<cmath>//引入fabs所需头文件

using namespace std;

int main()

{

double a,xn,xnp1;//定义a xn xn+1

cout << "请输入a的值:";//提示输入a的值

cin >> a;//输入a的值

xn = a;//将a的值赋给x

if (a >= 0)//如果a大于等于0 开方

{

for (; 1;)//死循环

{

xnp1 = (1.0 / 2) \* (xn + (a / xn));//给xn+1赋值

if (fabs(xnp1 - xn) < 0.00001)//fabs为绝对值 如果满足条件

{

cout << "a的平方根为" << xnp1;//输出当前xn+1(a的平方根)的值

break;//退出循环

}

xn = xnp1;//将此次xn+1的值赋给下次的xn

}

}

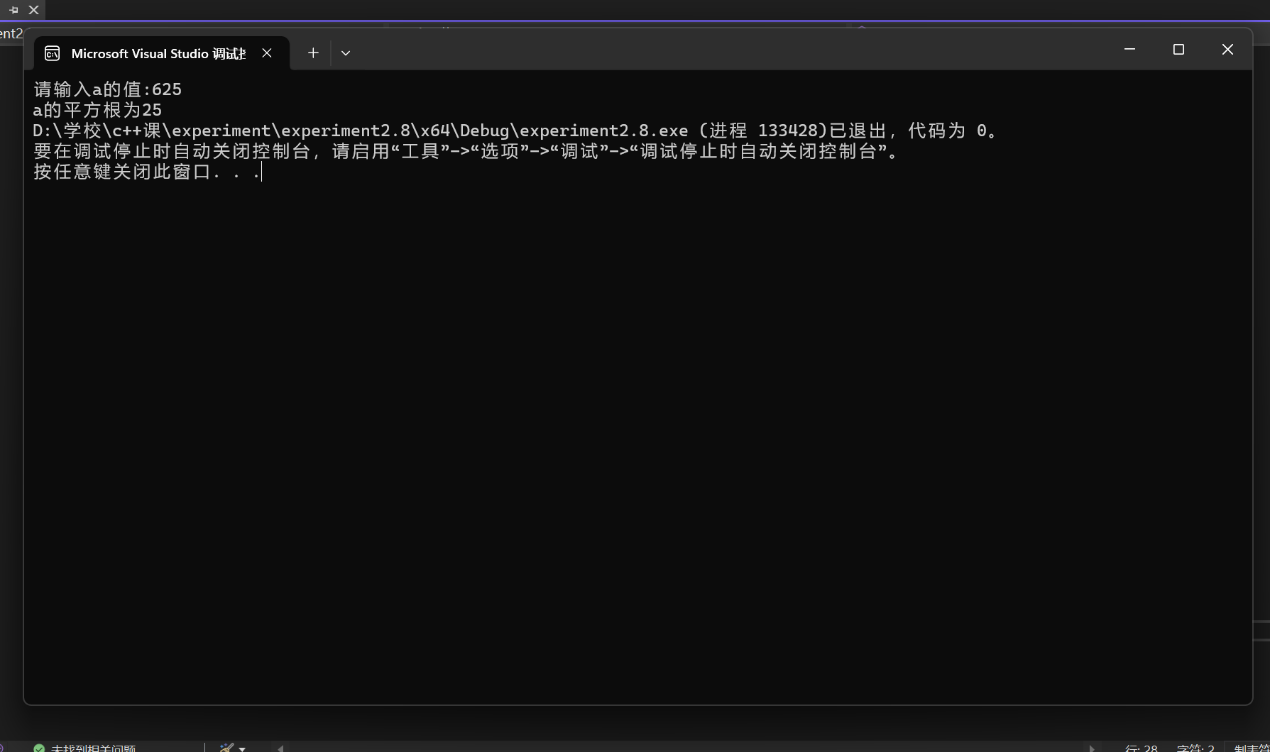
else//如果a小于0

{

cout << "开平方需要a>=0" << endl;//提示不能开方

}

return 0;

}

9.

#include<iostream>

using namespace std;

int main()

{

double price = 0.8, total\_p = 0;//定义单价及总价

int number = 2, day = 0;//定义购买数量及购买天数

double average\_p;//定义平均价格

for (int i = 1; number<=100 ; i++)//循环 number超过100退出循环

{

day++;//天数+1

total\_p += price \* number;//总价=原先总价+每天新花的钱

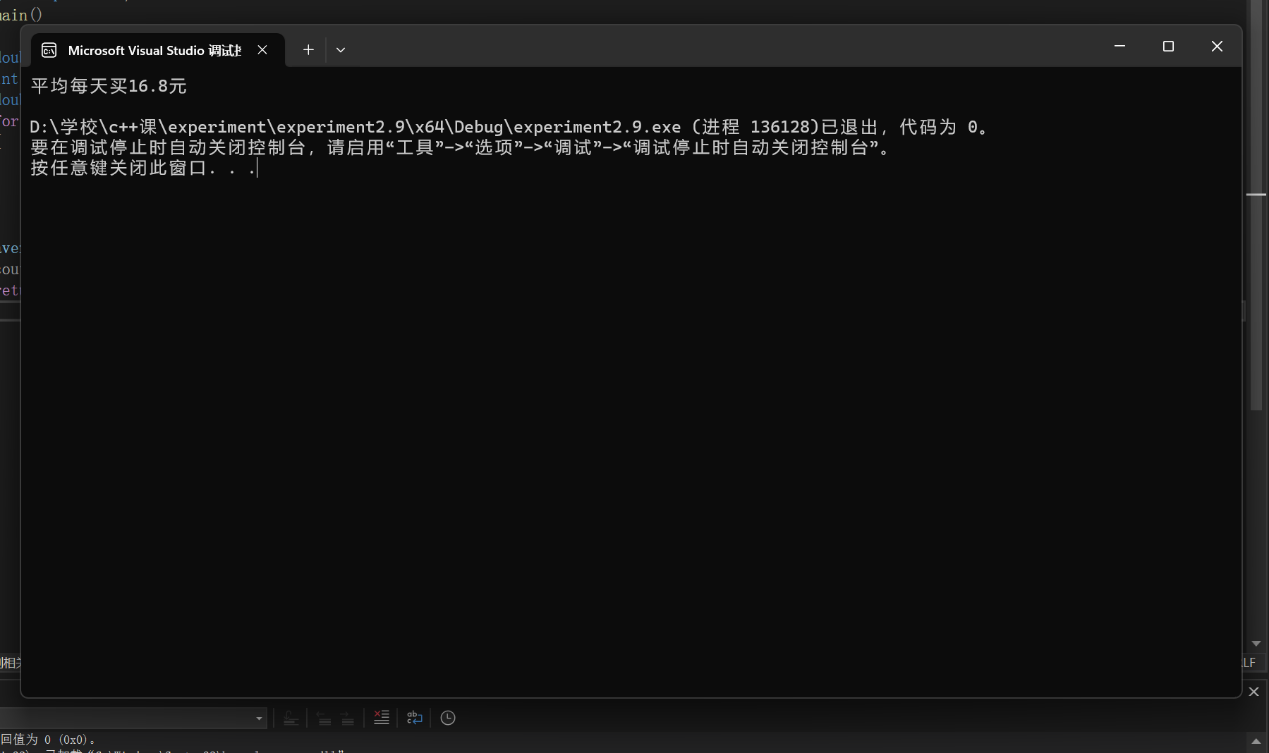
number \*= 2;//数量\*2

}

average\_p = total\_p / day;//平均价格=总价/天数

cout << "平均每天买" << average\_p << "元" << endl;//输出每天平均花费

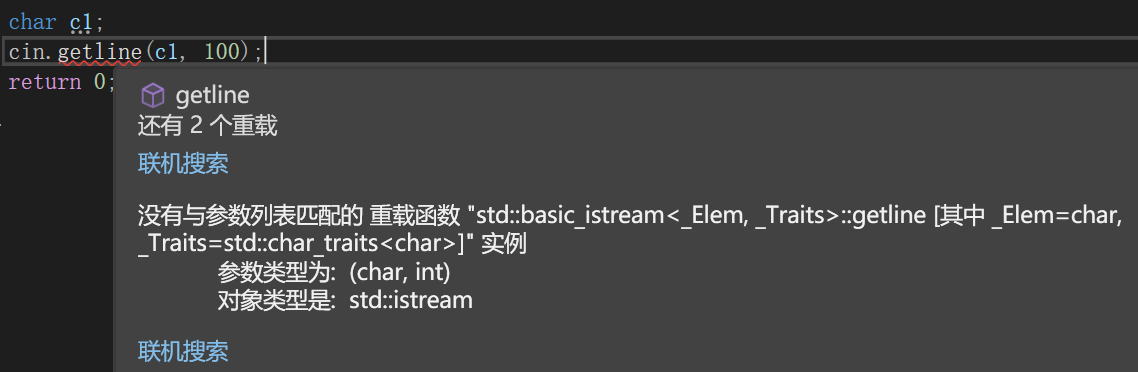
return 0;

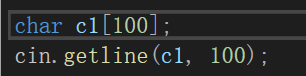
}

**四、遇到的问题与解决方法**

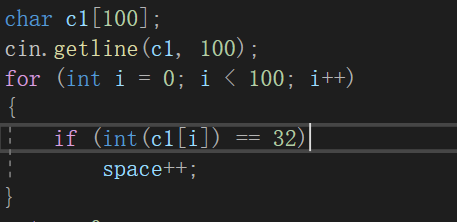
**1.不了解字符串中如何录入空格**

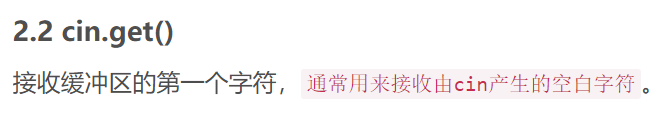
 最开始在互联网上查询相关资料，发现可以使用cin.getline()录入空格。

 在实际编程过程中，发现若想使用该函数，需要设置最大可接受字符量。在设置后，编译器报错。

 考虑到在使用该函数式时有定义最大可接受字符量，尝试将单个字符变量改编为字符变量类型数组后再次尝试，报错消除。

考虑到定义的是数组，若想要检测每个字符，需要引入循环，故写出循环。

**** 但是考虑到在此前已经将c1写成了数组，如果在最开始录入的过程中，没有录满我们所设定的100个元素，后面的没有被录入的数组，可能会干扰到最后的运行结果，再加上我目前没有掌握设立动态数组的知识，故只能放弃这一方法。

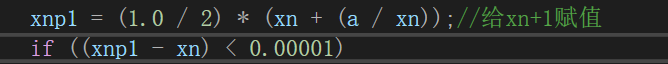
 再次在互联网上查找其他资料，发现cin.get（）也可以实现相关功能，故决定使用cin.get（）函数。

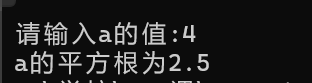
 但是在使用cin.get()函数时，因为不是很熟悉使用方法，发现在使用cin.get()录入后，再对结果进行输出，只能获得第一个字符。

 后通过询问其他同学，了解到要想逐个对cin.get()所录入的字符分类,需要将其写入循环当中。

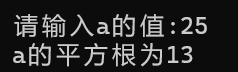
在这样写入后，程序能够对字符串中的每个字符进行判断，从而最终计算出各类型字符的数量。

**2.计算|xn+1-xn|时出现问题**

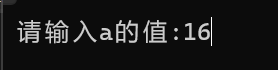
 在最开始计算|xn+1-xn|时，我的判断式写的是if（xnp1-xn<0.00001）。

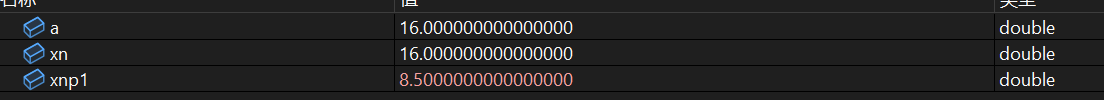
 但是在当时并没有注意到该错误，在直接运行程序后，发现结果不太准确。

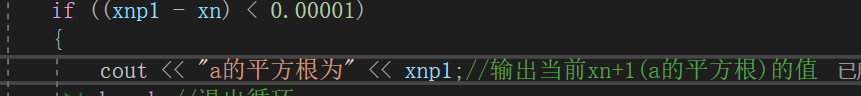
初步推测不准确有两种可能：（1）本算法本身带有一定误差；（2）程序某处出错。

 首先判断是否是由误差引起的不准确，分别计算25与16的平方根。

可见程序计算结果与其真实平方根相差极大，排除是误差引起的数据不准确。

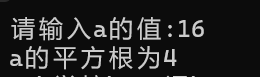
 接下来使用f11对程序进行逐步检查。在输入值时赋a=16。

 在进行if语句判断之前，各步骤计算均未出现错误。

 但是在各数据如上图所示时，if条件满足，直接执行了if条件里面的语句。

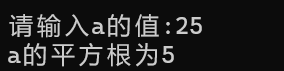
发现了是if的判断条件出错了，而要将if判断条件改正确，有两种方法：（1）(xnp1-xn<0.00001)&&(xn-xnp1<0.00001)（2）调用绝对值函数。

 首先尝试将判断条件修改为较为基础的逻辑表达式。

 程序可以正常运行。

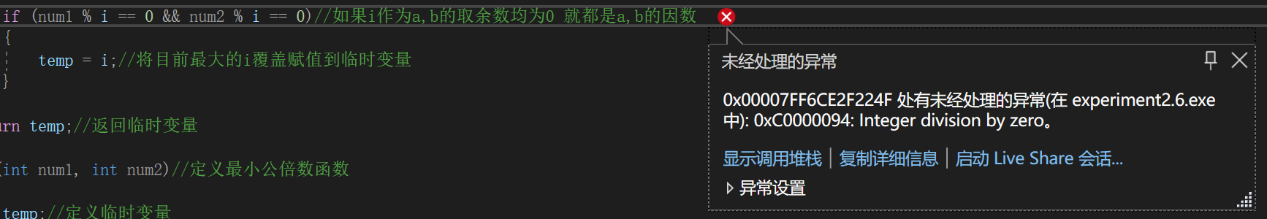
随后尝试通过调用绝对值的函数，来修改if的判断条件。

 在互联网上查询了相关资料后，发现可以使用fabs（）函数，要引入相对应的头文件。

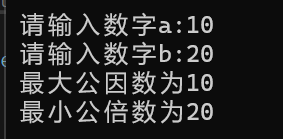
 在调用该函数后，发现可以正常计算，该方法也成立。

**3.计算最大公因数时报错**

计算最大公因数的思路是穷举法，从i=1时开始判断，每有一个符合公因数条件的值，就对公因数重新赋值，在最后i等于两数中的一个数后，停止穷举。

 在最开始设置i的初始值为0，遇到系统报错。

想起在之前测试时，遇到相似的报错是发生在除数等于0时或者取余数等于0时。

 发现报错是因为i的初始值设置错误，将i的初始值修改为1。

程序可以正常运行。

**五、体会**

**1.可以再对录入含空格的字符串进行更深的了解。**

在本次学习可以录入空格的函数过程中，我先尝试了cin.getline（）函数，后推测其可能需要和动态数组配合才能实现功能，但目前我认为我对动态数组了解较少，故放弃了使用该函数。

但是在微信群中，学长提示字符输入空格可以使用getline的方法，说明这个方法是可以进行尝试的。在下面一段时间我应该自己学习getline的相关用法，不管是否要配合动态数组使用，也把动态数组掌握了。

其次是，在改用了cin.get方法后，我对于这一方法的底层逻辑也不是很清楚，只知道把其写进循环就可以实现功能，我也应该学习这一方面的底层原理。

**2.在编写程序时要记得关注细节。**

在本次实验的过程中，我发现自己会对一些细节疏忽，从而造成一些小错误。例如在计算绝对值时忘记加上绝对值相关的判断逻辑或者忘记进行绝对值得运算，又或者说在计算类似a%i的循环时，将i的初始值设计为0导致程序报错。

其实这些细节的错误是可以避免的。在最后程序的运行阶段，如果发现程序存在细节问题，且编译器不报错，排查这些细节的过程将会变得十分麻烦且浪费时间。因此细节的关注十分有必要。